PORTAL

US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+exception +decompression

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used **exception** **decompression**

Found **100** of **5,782 searched out of 5,782.**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

☐ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 100

Result page: **1** 2 3 4 5 6 next

Relevance scale

1 A DISE implementation of dynamic code decompression

Marc L. Corliss, E. Christopher Lewis, Amir Roth

June 2003 **ACM SIGPLAN Notices , Proceedings of the 2003 ACM SIGPLAN conference on Language, compiler, and tool for embedded systems**, Volume 38 Issue 7

Full text available: pdf(291.52 KB) Additional Information: full citation, abstract, references, index terms

Code compression coupled with dynamic decompression is an important technique for both embedded and general-purpose microprocessors. *Post-fetch decompression*, in which decompression is performed after the compressed instructions have been fetched, allows the instruction cache to store compressed code but requires a highly efficient decompression implementation. We propose implementing post-fetch decompression using *dynamic instruction stream editing* (DISE), a programmable decoder-- ...

Keywords: DISE, code compression, code decompression

2 Grow & fold: compression of tetrahedral meshes

Andrzej Szymczak, Jarek Rossignac

June 1999 **Proceedings of the fifth ACM symposium on Solid modeling and applications**

Full text available: pdf(1.35 MB) Additional Information: full citation, references, citings, index terms

3 DISE: a programmable macro engine for customizing applications

Marc L. Corliss, E. Christopher Lewis, Amir Roth

May 2003 **ACM SIGARCH Computer Architecture News , Proceedings of the 30th annual international symposium on Computer architecture**, Volume 31 Issue 2

Full text available: pdf(335.30 KB) Additional Information: full citation, abstract, references

Dynamic Instruction Stream Editing (DISE) is a cooperative software-hardware scheme for efficiently adding customization functionality---e.g, safety/security checking, profiling, dynamic code decompression, and dynamic optimization---to an application. In DISE, application customization functions (ACFs) are formulated as rules for macro-expanding certain instructions into parameterized instruction sequences. The processor executes the rules on the fetched instructions, feeding the executi ...

4 Tool support for architectural decisions in embedded systems: CoCo: a hardware/software platform for rapid prototyping of code compression technologies

Haris Lekatsas, Jörg Henkel, Srimat Chakradhar, Venkata Jakkula, Murugan Sankaradass

June 2003 **Proceedings of the 40th conference on Design automation**

Full text available: pdf(243.90 KB) Additional Information: full citation, abstract, references, index terms

In recent years instruction code compression/decompression technologies have emerged as an efficient way to a) reduce the memory usage of an embedded system, b) to improve performance through effectively higher bandwidths and/or to c) reduce the overall power consumption of a system processing compressed code. We have presented efficient code compression/decompression techniques and architectures in the past. For the commercialization phase, we designed a novel hardware/software code compression ...

Keywords: code compression, embedded systems

5 DVI—a digital multimedia technology
G. David Ripley
July 1989 **Communications of the ACM**, Volume 32 Issue 7
Full text available: pdf(4.55 MB) Additional Information: full citation, abstract, references, citings, index terms, review

A digital presentation technology that manages anything from text to full-motion video has the potential of expanding the usefulness of personal computers, while rendering them less intimidating.

6 Techniques for obtaining high performance in Java programs
Iffat H. Kazi, Howard H. Chen, Berdenia Stanley, David J. Lilja
September 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 3
Full text available: pdf(816.13 KB) Additional Information: full citation, abstract, references, citings, index terms

This survey describes research directions in techniques to improve the performance of programs written in the Java programming language. The standard technique for Java execution is interpretation, which provides for extensive portability of programs. A Java interpreter dynamically executes Java bytecodes, which comprise the instruction set of the Java Virtual Machine (JVM). Execution time performance of Java programs can be improved through compilation, possibly at the expense of portabili ...

Keywords: Java, Java virtual machine, bytecode-to-source translators, direct compilers, dynamic compilation, interpreters, just-in-time compilers

7 Static load classification for improving the value predictability of data-cache misses
Martin Burtscher, Amer Diwan, Matthias Hauswirth
May 2002 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2002 Conference on Programming language design and implementation**, Volume 37 Issue 5
Full text available: pdf(273.26 KB) Additional Information: full citation, abstract, references, index terms

While caches are effective at avoiding most main-memory accesses, the few remaining memory references are still expensive. Even one cache miss per one hundred accesses can double a program's execution time. To better tolerate the data-cache miss latency, architects have proposed various speculation mechanisms, including load-value prediction. A load-value predictor guesses the result of a load so that the dependent instructions can immediately proceed without having to wait for the memory access ...

Keywords: load-value prediction, type-based analysis

8 Java bytecode compression for low-end embedded systems
Lars Ræder Clausen, Ulrik Pagh Schultz, Charles Consel, Gilles Muller
May 2000 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 22 Issue 3
Full text available: pdf(241.04 KB) Additional Information: full citation, abstract, references, citings, index terms, review

A program executing on a low-end embedded system, such as a smart-card, faces scarce memory resources and fixed execution time constraints. We demonstrate that factorization of common instruction sequences in Java bytecode allows the memory footprint to be reduced, on average, to 85% of its original size, with a minimal execution time penalty. While preserving Java compatibility, our solution requires only a few modifications which are straightforward to implement in any JVM used in a low-e ...

Keywords: Java bytecode, code compression, embedded systems

9 The SimpleScalar tool set, version 2.0
Doug Burger, Todd M. Austin
June 1997 **ACM SIGARCH Computer Architecture News**, Volume 25 Issue 3
Full text available: pdf(985.46 KB) Additional Information: full citation, abstract, citings, index terms

This document describes release 2.0 of the SimpleScalar tool set, a suite of free, publicly available simulation tools that offer both detailed and high-performance simulation of modern microprocessors. The new release offers more tools and capabilities, precompiled binaries, cleaner interfaces, better documentation, easier installation, improved portability, and higher performance. This paper contains a complete description of the tool set, including retrieval and installation instructions, a d ...

10 Compilation and run-time systems: DELI: a new run-time control point
Giuseppe Desoli, Nikolay Mateev, Evelyn Duesterwald, Paolo Faraboschi, Joseph A. Fisher
November 2002 **Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture**
Full text available: pdf(1.27 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

The Dynamic Execution Layer Interface (DELI) offers the following unique capability: it provides fine-grain control over the execution of programs, by allowing its clients to observe and optionally manipulate every single instruction---at run time---just before it runs. DELI accomplishes this by opening up an interface to the layer between the execution of software and hardware. To avoid the slowdown, DELI caches a private copy of the executed code and always runs out of its own private cache.In ...

11 Test data compression using dictionaries with selective entries and fixed-length indices
Lei Li, Krishnendu Chakrabarty, Nur A. Touba
October 2003 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 8 Issue 4
Full text available: pdf(594.46 KB) Additional Information: full citation, abstract, references, index terms

We present a dictionary-based test data compression approach for reducing test data volume in SOCs. The proposed method is based on the use of a small number of ATE channels to deliver compressed test patterns from the tester to the chip and to drive a large number of internal scan chains in the circuit under test. Therefore, it is especially suitable for a reduced pin-count and low-cost DFT test environment, where a narrow interface between the tester and the SOC is desirable. The dictionary-ba ...

Keywords: Embedded core testing, SoC testing, reduced pin-count testing, test application time, test data volume

12 Compiler-driven cached code compression schemes for embedded ILP processors
Sergei Y. Larin, Thomas M. Conte
November 1999 **Proceedings of the 32nd annual ACM/IEEE international symposium on Microarchitecture**
Full text available: pdf(1.24 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

During the last 15 years, embedded systems have grown in complexity and performance to rival desktop systems. The architectures of these systems present unique challenges to processor microarchitecture, including instruction encoding and instruction fetch processes. This paper presents new techniques for reducing embedded system code size without reducing functionality. This approach is to extract the pipeline decoder logic for an embedded VLIW processor in software at system develo ...

13 The ULTRAVIS system
Gunter Knittel
October 2000 **Proceedings of the 2000 IEEE symposium on Volume visualization**
Full text available: pdf(428.76 KB) Additional Information: full citation, references, citings, index terms

Keywords: raycasting, volume rendering

14 Computing curricula 2001
September 2001 **Journal on Educational Resources in Computing (JERIC)**
Full text available: pdf(613.63 KB) html(2.78 KB) Additional Information: full citation, references, citings, index terms

15 Multimedia for tiny devices: Integrated power management for video streaming to mobile handheld devices
Shivajit Mohapatra, Radu Cornea, Nikil Dutt, Alex Nicolau, Nalini Venkatasubramanian
November 2003 **Proceedings of the eleventh ACM international conference on Multimedia**
Full text available: pdf(417.95 KB) Additional Information: full citation, abstract, references, index terms

Optimizing user experience for streaming video applications on handheld devices is a significant research challenge. In this paper, we propose an integrated power management approach that unifies low level architectural optimizations (CPU, memory, register), OS power-saving mechanisms (Dynamic Voltage Scaling) and adaptive middleware techniques (admission control, optimal transcoding, network traffic regulation). Specifically, we identify interaction parameters between the different levels and o ...

Keywords: cross-layer adaptation, low-power, multimedia streaming

16 Increasing effective IPC by exploiting distant parallelism
Iván Martel, Daniel Ortega, Eduard Ayguadé, Mateo Valero
May 1999 **Proceedings of the 13th international conference on Supercomputing**
Full text available: pdf(1.20 MB) Additional Information: full citation, references, index terms

17 Special session on memory wall: Dynamic techniques to reduce memory traffic in embedded systems
Ben Juurlink, Pepijn de Langen
April 2004 **Proceedings of the first conference on computing frontiers on Computing frontiers**
Full text available: pdf(266.88 KB) Additional Information: full citation, abstract, references, index terms

Memory transfers, in particular from/to off-chip memories, consume a significant amount of power. In order to reduce the amount of off-chip memory traffic, one or more levels of cache can be employed, located on the same die as the processor core. For performance, energy, and cost reasons, it is expedient that the on-chip cache is small and direct-mapped. Small, direct-mapped caches, however, generally produce much more traffic than needed. The purpose of this paper is two-fold. First, to measur ...

Keywords: caches, embedded processors, memory traffic, power consumption

18 Understanding and improving operating system effects in control flow prediction
Tao Li, Lizy Kurian John, Anand Sivasubramaniam, N. Vijaykrishnan, Juan Rubio
October 2002 **Tenth international conference on architectural support for programming languages and operating systems on Proceedings of the 10th international conference on architectural support for programming languages and operating systems (ASPLOS-X)**, Volume 37 , 30 , 36 Issue 10 , 5 , 5
Full text available: pdf(1.50 MB) Additional Information: full citation, abstract, references, citings

Many modern applications result in a significant operating system (OS) component. The OS component has several implications including affecting the control flow transfer in the execution environment. This paper focuses on understanding the operating system effects on control flow transfer and prediction, and designing architectural support to alleviate the bottlenecks. We characterize the control flow transfer of several emerging applications on a commercial operating system. We find that the ex ...

19 Profile-based dynamic voltage and frequency scaling for a multiple clock domain microprocessor
Grigorios Magklis, Michael L. Scott, Greg Semeraro, David H. Albonesi, Steven Dropsho
May 2003 **ACM SIGARCH Computer Architecture News , Proceedings of the 30th annual international symposium on Computer architecture**, Volume 31 Issue 2
Full text available: pdf(541.83 KB) Additional Information: full citation, abstract, references

A Multiple Clock Domain (MCD) processor addresses the challenges of clock distribution and power dissipation by dividing a chip into several (coarse-grained) clock domains, allowing frequency and voltage to be reduced in domains that are not currently on the application's critical path. Given a reconfiguration mechanism capable of choosing appropriate times and values for voltage/frequency scaling, an MCD processor has the potential to achieve significant energy savings with low performance degr ...

20 Code compression: Reducing code size with echo instructions
Jeremy Lau, Stefan Schoenmackers, Timothy Sherwood, Brad Calder
October 2003 **Proceedings of the international conference on Compilers, architectures and synthesis for embedded systems**
Full text available: pdf(228.46 KB) Additional Information: full citation, abstract, references, index terms

In an embedded system, the cost of storing a program on-chip can be as high as the cost of a microprocessor. Compressing an application's code to reduce the amount of memory required is an attractive way to decrease costs. In this paper, we examine an executable form of program compression using *echo* instructions.With echo instructions, two or more similar, but not necessarily identical, sections of code can be reduced to a single copy of the repeating code. The single copy is left in the ...

Keywords: code size optimization, compression, echo instructions

Results 1 - 20 of 100 Result page: **1** 2 3 4 5 6 next